

# Uso de redes neuronales pulsantes para mejorar el filtrado de imágenes contaminadas con ruido Gaussiano

Estela Ortiz Rangel, Manuel Mejía-Lavalle, Humberto Sossa Azuela

Centro Nacional de Investigación y Desarrollo Tecnológico,  
Departamento de Ciencias Computacionales, Cuernavaca, Morelos,  
México

Instituto Politécnico Nacional, CIC, Ciudad de México  
México

{estela\_or, mlavalle} @cenidet.edu.mx, hsossa@cic.ipn.mx

**Resumen.** Se propone un algoritmo llamado ICM-TM para reducir el efecto de ruido gaussiano en imágenes en escala de grises basado en una Red Neuronal Artificial tipo Pulso-Acoplada simplificada llamada Intersección Cortical Model (ICM). Una matriz de tiempos (TM) concentra la información respectiva al número de iteración donde se activa por primera vez la neurona correspondiente a cada pixel; basándose en los tiempos de activación de las neuronas se establece un criterio de filtrado selectivo combinando los operadores mediana y promedio. El desempeño del algoritmo propuesto fue evaluado experimentalmente, con distintos grados de ruido gaussiano y los resultados de las simulaciones muestran que la efectividad del método es superior al filtro de mediana convencional, al filtro Wiener y a la técnica Pulse-Coupled Neural Networks with the Null Interconnections (PCNNNI); los resultados están representados por el parámetro Peak Signal to Noise Ratio (PSNR) principalmente.

**Palabras clave:** Intersection Cortical Model (ICM), ruido gaussiano, filtro Wiener, Peak Signal to Noise Ratio (PSNR).

## Using Pulsed Neural Networks to Improve Filtering of Images Contaminated with Gaussian Noise

**Abstract.** An algorithm called ICM-TM is proposed to reduce the effect of Gaussian noise in grayscale images based on a kind of Artificial Neural Networks type Pulse-Coupled simplified called Intersection Cortical Model (ICM). A Time Matrix (TM) provides the respective information to the iteration number where first activated neuron corresponding to each pixel; a selective filtering criteria is established combining the median and average operators based on the neurons activation times. The performance of the proposed algorithm was evaluated experimentally with varying degrees of Gaussian noise and the results of the simulations show that the effectiveness of the method is superior to the median filter, Wiener filter and to the Pulse-Coupled Neural Networks with the Null

Interconnections (PCNNNI); the results are mainly represented by the parameter Peak Signal to Noise Ratio (PSNR).

**Keywords:** Intersection cortical model (ICM), Gaussian noise, Wiener filter, peak signal to noise ratio (PSNR).

## 1. Introducción

El ruido en las imágenes digitales es información no deseada que las modifica; puede deberse a defectos en los dispositivos de captura, en los medios de transmisión o almacenamiento y causa problemas para el procesamiento posterior de las imágenes.

El ruido aditivo gaussiano  $q$  es un modelo que simula la afectación aleatoria de todos los píxeles de una imagen con valores uniformemente distribuidos, donde su función de densidad de probabilidad  $p_q(x)$  está dada en términos del promedio  $\mu$  y la varianza  $\sigma^2$  de una variable aleatoria  $x$  como se expresa en la ec. 1 [1]. Este tipo de ruido es muy común en las imágenes digitales y debido a sus características es difícil de eliminar por completo.

$$p_q(x) = (2\pi\sigma^2)^{-1/2} e^{-(x-\mu)^2/2\sigma^2}. \quad (1)$$

Existen métodos para la reducción del ruido gaussiano que pueden ser clasificados como filtros espaciales y frecuenciales. Los filtros espaciales pueden ser lineales tales como el filtro promedio y el gaussiano y los no lineales como el filtro de mediana y el filtro sigma.

El filtro *Wiener*  $H(u, v)$  como se expresa en la ec. 2 [2] es un filtro frecuencial que se basa en la reducción del error cuadrático medio y mejora sustancialmente la calidad de una imagen con ruido gaussiano, sin embargo requiere del cálculo del espectro de energía de la imagen ideal  $S_w(u, v)$  y del ruido  $S_f(u, v)$ , de la estimación de una función de degradación  $D(u, v)$  y su conjugado  $D^*(u, v)$  para realizar el filtrado.

$$H(u, v) = \frac{D^*(u, v)}{D^*(u, v)D(u, v) + \frac{S_w(u, v)}{S_f(u, v)}}. \quad (2)$$

Algunas modificaciones han sido propuestas a este filtro con el fin de mejorar su desempeño y generalizar su aplicación [3]. El filtro *Sigma* también es uno de los más efectivos en la eliminación del ruido gaussiano y es ampliamente utilizado por su simplicidad, no obstante su capacidad de preservar bordes aún se sigue mejorando, por ejemplo por medio de técnicas difusas [4].

También se ha propuesto técnicas que utilizan la información de los bordes de la imagen contaminada, donde se aplican los principios de similaridad, no obstante su proceso computacional es largo [5, 6] y por otro lado están las técnicas basadas en transformadas *wavelet* son promisorias pero conllevan largos procesos de cálculo, lo cual dificulta su implementación en tiempo real y en sistemas embebidos [7].

Una forma distinta de enfrentar el problema de eliminación del ruido gaussiano en imágenes digitales ha surgido de la exploración experimental, tal es el caso de las Redes

Neuronales Artificiales de tercera generación llamadas Redes Neuronales Pulso-Acopladas o por sus siglas en inglés PCNN (*Pulse Coupled Neural Networks*), las cuales han sido empleadas de modo eficiente para el procesamiento de imágenes en diversas tareas como la segmentación, la clasificación, la identificación de imágenes, entre otras [8].

Las redes tipo PCNN son un modelo matemático, propuesto por Eckhorn, basado en la frecuencia de activación de las neuronas de la corteza visual de los mamíferos [9]. El tiempo y frecuencia de activación de las neuronas han sido utilizados para procesar imágenes gracias al modelo computacional simplificado de PCNN propuesto por Ranganath y Kuntimad [10].

Las propiedades del modelo de PCNN permiten a cada neurona corresponder con un pixel y relacionar su nivel de gris con el de sus vecinos, de modo que al iterar la Red Neuronal el tiempo de activación de cada neurona se registra en una matriz. Esta información puede ser utilizada para la detección de los pixeles con ruido y la aplicación de una técnica selectiva de filtrado de ruido gaussiano [11, 12], e incluso para la reducción del ruido gaussiano y de sal y pimienta mezclados [13, 14].

Diversos modelos simplificados de PCNN han sido propuestos para trabajar computacionalmente con imágenes; dos de las principales variaciones son *Intersection Cortical Model (ICM)* y *Pulse-Coupled Neural Networks with the Null Interconnections (PCNNNI)* [10]. Ma [15] ha utilizado estos métodos combinándolos con el operador de mediana, promedio y morfológico o con el filtro *Wiener* y la matriz de tiempos para reducir el ruido de Sal y Pimienta y el ruido gaussiano en imágenes digitales.

El método ICM-TM que se propone consiste en utilizar una red ICM para generar la matriz de tiempos y aplicar selectivamente los operadores mediana y promedio para suprimir el ruido gaussiano en imágenes digitales en escala de grises, de forma que se logre una significativa reducción del ruido con un método computacional simple que puede ser implementado para aplicaciones con sistemas embebidos.

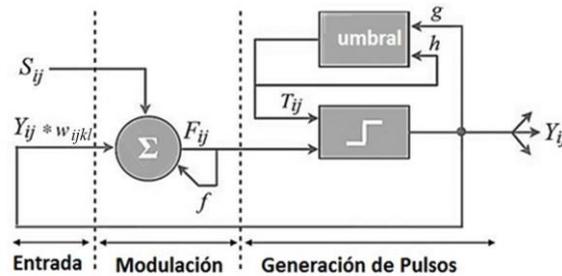
## 2. Modelo de PCNN e ICM

Las Redes Neuronales Pulso-Acopladas son un sistema que emula efectivamente a las neuronas biológicas de la corteza visual de los mamíferos y han sido aplicadas en variedad de dominios, especialmente en el procesamiento de imágenes han sido útiles para remoción de ruido, reconocimiento de objetos, optimización, adelgazamiento, segmentación, fusión, identificación y remoción de sombras [8,11].

En el ámbito del procesamiento de imágenes las diferencias entre las redes Neuronales Artificiales tradicionales y las Redes Neuronales de tercera generación son evidentes tanto en su configuración como en su operación.

Las redes tipo PCNN no requieren entrenamiento ya que su función es clasificar los pixeles por sus niveles de intensidad, en este modelo cada pixel de la imagen corresponde a una neurona y su valor es introducido a la red por medio de una señal llamada *feeding*. El umbral de activación de las neuronas es dinámico, además cada neurona recibe información de sus vecinas a través de una sinapsis, lo cual se conoce como *linking*. *Feeding* y *linking* se combinan para formar el potencial interno de la neurona, que al ser comparado con el umbral produce salidas binarias. Estas

características hacen que las neuronas correspondientes a pixeles vecinos con valores de intensidad similares se activen al mismo tiempo en ciertas regiones, a lo que se denomina activación de pulsos síncrona [16].



**Fig. 1.** Diagrama general de ICM [14]

El modelo original de PCNN tiene algunas limitantes en la práctica debido al gran número de interconexiones entre neuronas y al establecimiento de parámetros de operación; por este motivo fue obtenido un modelo iterativo simplificado de PCNN que permitiera extender el uso de estas redes en el procesamiento de imágenes.

ICM es un tipo de PCNN más simple (Fig. 1), donde no se considera *linking* en las neuronas y donde el *feeding*  $F_{ij}$  mantiene su salida con un factor de decaimiento dado por  $f$ . La señal  $F_{ij}$  está compuesta por la última salida del vecindario de neuronas  $Y_{ij}$  ponderada por una matriz  $w_{ijkl}$  (generalmente gaussiana) y por el estímulo externo de entrada  $S_{ij}$  (nivel de gris de cada pixel normalizado entre 0 y 1). El umbral dinámico para cada pixel  $T_{ij}$  crece obedeciendo a  $h$  cuando su salida se activa y mantiene su estado previo con una atenuación dada por  $g$ , lo cual da origen a la formación de pulsos.

El modelo computacional iterativo ICM puede ser descrito por medio de las siguientes funciones (3)-(5) [15]:

$$F_{ij}[n] = fF_{ij}[n-1] + \sum w_{ijkl}Y_{kl}[n-1] + S_{ij}, \quad (3)$$

$$T_{ij}[n] = gT_{ij}[n-1] + hY_{ij}[n-1], \quad (1)$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } F_{ij}[n] > T_{ij}[n] \\ 0 & \text{en otro caso} \end{cases} \quad (2)$$

donde  $n$  es la iteración actual,  $w_{ijkl}$  es la matriz de pesos sinápticos que liga una neurona con sus vecinas y finalmente  $f, g$  y  $h$  son coeficientes de ajuste, donde típicamente  $g < 1.0$ ,  $f < g$  y  $h$  es un valor grande.

### 3. La matriz de tiempos (TM)

Las redes tipo PCNN pueden ser utilizadas para determinar la posición de los pixeles ruidosos con base en las neuronas activadas a la salida de la red por iteración y, al aplicar un operador de mediana que elimine los valores más altos y más bajos, es posible eliminar el ruido de Sal y Pimienta.

No obstante para el filtrado del ruido gaussiano es necesario emplear otra técnica ya que todos los píxeles han sido afectados en algún grado, por lo que se introduce la matriz de tiempos de las mismas dimensiones que la imagen a tratar; ella contiene información relacionada con la estructura espacial de la imagen con ruido, es decir, realiza un mapeo de la información espacial en una secuencia temporal [11].

La matriz de tiempos obtenida a partir de una red neuronal con interconexión nula PCNNNI [15] ha sido utilizada para detectar los píxeles ruidosos y procesada para reducir el ruido gaussiano combinándola con otros métodos [12]. El modelo PCNNNI no considera la señal de *linking* y elimina la influencia de la matriz de pesos sinápticos que liga a una neurona con sus vecinas, por lo que el potencial interno de una neurona, su umbral y salida únicamente dependen de la intensidad de su píxel correspondiente, además de que dicha técnica modifica el rango dinámico de la imagen para reducir el número de iteraciones necesarias y aplica cinco criterios de filtrado [15].

El método propuesto ICM-TM consiste en obtener la matriz de tiempos conservando la información de la relación espacial entre píxeles, de modo que las diferencias en la intensidad de los píxeles de la imagen originen diferencias en la secuencia de activación de sus respectivas neuronas. Para conservar el tiempo de activación de cada neurona se define una matriz  $M_{ij}$  que puede ser descrita como [11]:

$$M_{ij}[n] = \begin{cases} n & \text{si } Y_{ij}[n] = 1 \\ M_{ij}[n - 1] & \text{en otro caso} \end{cases} \quad (6)$$

El algoritmo ICM-TM conserva el rango dinámico de la imagen y complementa el proceso de filtrado obteniendo la información de la matriz de tiempos para cualquier número de iteraciones, suavizando regiones de 5x5 píxeles con el filtro promedio para suavizar la imagen o bien conservando información de la imagen pero suprimiendo valores extremos mediante el operador de mediana.

#### 4. Descripción del algoritmo propuesto ICM-TM

En nuestro método todas las neuronas de ICM están ligadas mutuamente del mismo modo y sus salidas tienen dos estados posibles, activado y no activado. Este modelo es más rápido que PCNN debido a que implica menos ecuaciones y es más adaptable ya que tiene menos parámetros que ajustar.

El proceso principal del algoritmo ICM-TM (Fig. 2) es como sigue:

*Paso 1.* Pasar los valores de gris de los píxeles de la imagen con ruido  $O_{ij}$ , normalizados entre 0 y 1 a la red ICM por medio de  $S_{ij}$ .

*Paso 2.* Cuando una neurona en la posición  $(i, j)$  tome el valor de activada por primera vez ( $Y_{ij}[n] = 1$ ) registrar el número de iteración actual  $n$  en la matriz de tiempos  $M_{ij}[n]$  en el lugar correspondiente al píxel procesado  $(i, j)$ , omitiendo la primera iteración.

*Paso 3.* Continuar iterando la red hasta que todos los píxeles se hayan activado por lo menos una vez, es decir cuando todos los elementos en  $M_{ij}[n]$  son distintos de cero.

*Paso 4.* Con base en la matriz de tiempos  $M_{ij}[n]$  aplicar el siguiente criterio:

a) Si el tiempo de activación de una neurona es igual a la mediana de los tiempos en una vecindad de 25 neuronas centradas en ésta, el valor del pixel en la imagen filtrada será el valor truncado en entero resultante de aplicar el operador promedio sobre esta vecindad en la imagen con ruido  $O_{ij}$ .

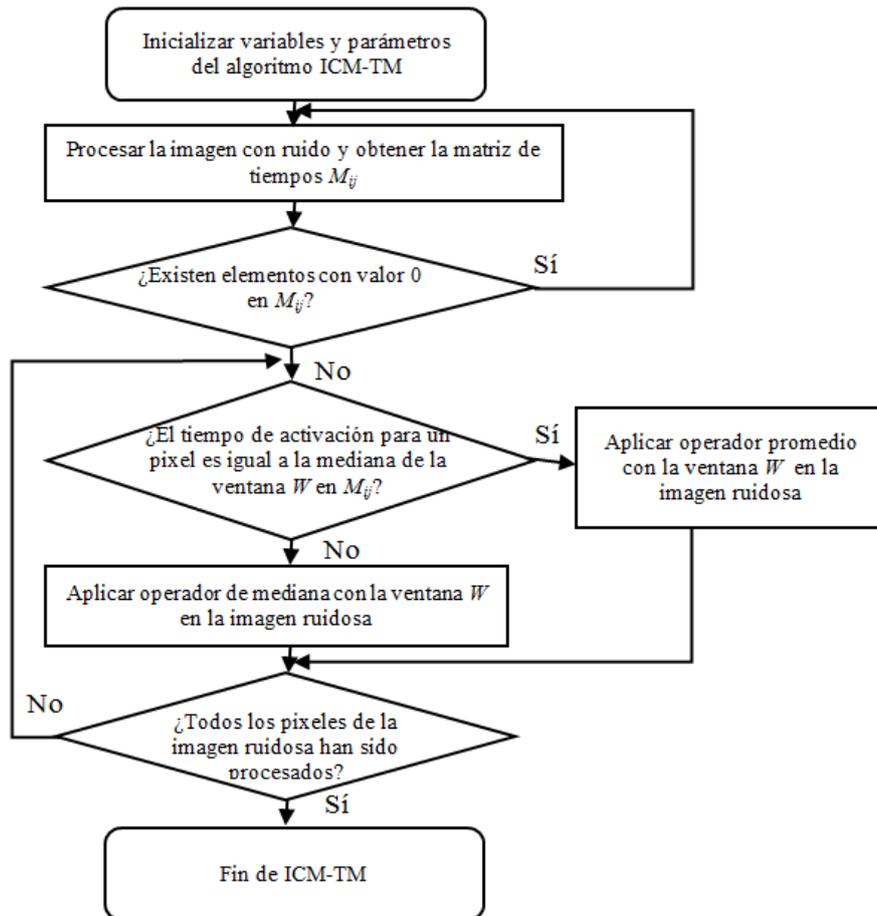


Fig. 2. Diagrama del algoritmo propuesto

b) Si no el valor del pixel de la imagen filtrada será el resultante de aplicar el operador de mediana sobre una vecindad de 25 pixeles centrada en este pixel de la imagen con ruido  $O_{ij}$ . En este caso la ventana  $W$  es de 5x5, seleccionada para un mejor efecto de suavizado.

## 5. Experimentación y resultados

Las pruebas experimentales fueron realizadas con las imágenes en escala de grises de 255 valores: *Lena*, *Peppers* y *Baboon*; con tamaño de 512x512 pixeles.

Durante el proceso de experimentación, se introdujo a las imágenes ruido gaussiano de promedio 0 y varianza entre 0.01 y 0.09. La imagen  $O_{ij}$  normalizada entre 0 y 1 se ingresó al ICM como  $S_{ij}[n]$ . Las matrices  $F_{ij}[n]$ ,  $Y_{ij}[n]$ ,  $T_{ij}[n]$  y  $M_{ij}[n]$  son de las mismas dimensiones que la imagen y fueron inicializadas en 0.

Los otros parámetros fueron empíricamente seleccionados para simulación como:

- La matriz interna de pesos formada con valores gaussianos en función de la distancia:

$$w_{ijkl} = \begin{bmatrix} 0.5 & 1 & 0.5 \\ 1 & 0 & 1 \\ 0.5 & 1 & 0.5 \end{bmatrix}.$$

- Constantes:  $f = 0.9, g = 0.8, h = 20$ .
- En el paso 4, el operador de mediana y el operador promedio están definidos de acuerdo a lo siguiente: Si  $x_{ij}$  denota al pixel con coordenadas  $(i, j)$  en la imagen con ruido y  $X_{ij}$  denota el conjunto de pixeles en la ventana  $W$  con el vecindario  $(2K + 1) \times (2K + 1)$  centrada en  $x_{ij}$ , entonces:

$$X_{ij} = \{x_{i-K,j-K}, \dots, x_{ij}, \dots, x_{i+K,j+K}\}. \quad (7)$$

- La mediana de la ventana de la imagen está definida como

$$m_{ij} = \text{mediana}(X_{ij}). \quad (8)$$

- El promedio de la ventana de la imagen está definido como

$$p_{ij} = \text{promedio}(X_{ij}). \quad (9)$$

El algoritmo ICM-TM se comparó contra dos filtros clásicos, el filtro de mediana de 3x3 y el filtro *Wiener* de 3x3, y contra el PCNNNI basado en el algoritmo de Ma [15].

Se realizó la medición de los resultados obtenidos con tres métricas, las cuales están formuladas como sigue [13]:

a) *PSNR* (*Peak Signal to Noise Ratio*, dB), el cual es utilizado para medir la habilidad de supresión del ruido, mientras más grande es su valor mejor es el efecto del filtrado,

$$PSNR = 10 \log_{10} \left( \frac{f(m,n)^2}{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [f(m,n) - f'(m,n)]^2} \right) [dB]. \quad (10)$$

b) *MAE* (*Mean Absolute Error*), indica la calidad del filtrado como la preservación de detalles finos, para lo que cual de ser minimizado,

$$MAE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |f(m,n) - f'(m,n)|, \quad (11)$$

donde  $M$  y  $N$  denotan las filas y columnas de la imagen,  $f(m, n)$  denota la imagen sin ruido y  $f'(m, n)$  es la imagen resultante del proceso de filtrado.

c) *NMSE (Normalized Mean Square Error)*, un mejor método de filtrado debe generar un menor valor resultante de *NMSE*.

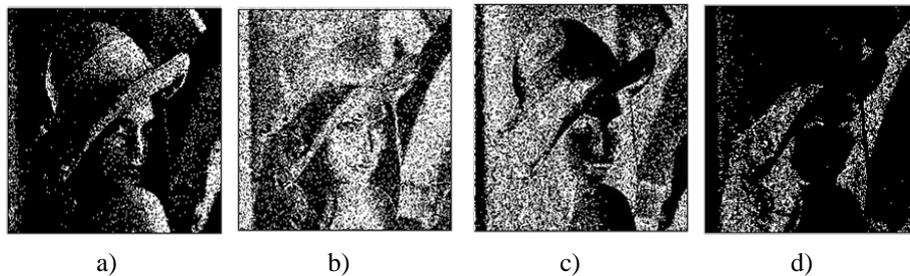
$$NMSE = \frac{\sum_{m=1}^M \sum_{n=1}^N [f(m, n) - f'(m, n)]^2}{\sum_{m=1}^M \sum_{n=1}^N [f(m, n)]^2}. \quad (12)$$

De manera adicional también se midió el tiempo de cómputo, el cual se realizó bajo el sistema operativo Windows 7 Ultimate SP1, con un procesador Intel Core i7 a 3.40 GHz y 8GB de RAM utilizando Matlab.

## 6. Desempeño del método propuesto

Para ICM cada salida es distinta en cada iteración y la activación de las neuronas obedece a la relación entre los niveles de gris de la imagen con ruido gaussiano de promedio 0 y varianza 0.01 (Fig. 3). La red puede iterarse cuantas veces se desee dependiendo del objetivo que se persiga. Para el caso de filtrado de ruido gaussiano se requiere encontrar información sobre el grado de contaminación de cada pixel, por lo que se analizó la matriz de tiempos.

La información de la matriz de tiempos es numérica, por lo que se pueden distinguir los pixeles que se activan por primera vez en cada iteración y se extraen los pixeles que guardan mayor información de la imagen, discriminando los que no para eliminar la mayor cantidad de ruido no deseado en la imagen.



**Fig. 3.** Salidas binarias de la red ICM para las iteraciones a) 5, b) 6, c) 7 y d) 8

Para dar un ejemplo numérico del procedimiento de filtrado considérese una matriz correspondiente a una imagen en escala de grises contaminada con ruido gaussiano de promedio 0 y varianza 0.08 de la que se tomó una ventana de 6 x6 pixeles  $O_{ij}$  y su correspondiente imagen normalizada  $S_{ij}$ :

$$O_{ij} = \begin{pmatrix} 238 & 139 & 255 & 239 & 112 & 182 \\ 68 & 235 & 59 & 255 & 178 & 42 \\ 179 & 52 & 138 & 227 & 116 & 81 \\ 238 & 45 & 114 & 67 & 205 & 51 \\ 147 & 54 & 148 & 128 & 37 & 255 \\ 180 & 0 & 141 & 25 & 110 & 56 \end{pmatrix}, S_{ij} = \begin{pmatrix} 0.9333 & 0.5451 & 1.0000 & 0.9373 & 0.4392 & 0.7137 \\ 0.2667 & 0.9216 & 0.2314 & 1.0000 & 0.6980 & 0.1647 \\ 0.7020 & 0.2039 & 0.5412 & 0.8902 & 0.4549 & 0.3176 \\ 0.9333 & 0.1765 & 0.4471 & 0.2627 & 0.8039 & 0.2000 \\ 0.5765 & 0.2118 & 0.5804 & 0.5020 & 0.1451 & 1.0000 \\ 0.7059 & 0 & 0.5529 & 0.0980 & 0.4314 & 0.2196 \end{pmatrix}$$

Se introduce la imagen normalizada en la red ICM y se itera 10 veces, obteniéndose el umbral dinámico de dicha iteración  $T_{ij}$ , los valores en  $x$  son indistintos para este ejemplo, tal es el caso de los bordes de la imagen:

$$T_{ij} = \begin{pmatrix} 16.1554 & 19.3554 & 16.1554 & 16.1554 & 19.3554 & x \\ 19.3554 & \underline{13.5954} & 16.1554 & \underline{13.5954} & 16.1554 & x \\ 19.3554 & 19.3554 & 16.1554 & 16.1554 & 19.3554 & x \\ 16.1554 & 19.3554 & 19.3554 & 19.3554 & 16.1554 & x \\ 19.3554 & 23.3554 & 19.3554 & 19.3554 & 23.3554 & x \\ x & x & x & x & x & x \end{pmatrix}.$$

Luego se obtiene el potencial interno de la matriz  $F_{ij}$ :

$$F_{ij} = \begin{pmatrix} 9.3197 & 8.4762 & 11.4841 & 11.0754 & 5.9214 & x \\ 6.7977 & \underline{13.7602} & 8.8778 & \underline{14.0460} & 9.6073 & x \\ 9.7184 & 8.9600 & 11.1566 & 13.3848 & 7.9292 & x \\ 11.4509 & 9.0972 & 10.8096 & 9.5191 & 10.6130 & x \\ 7.0908 & 6.6561 & 9.2021 & 8.6462 & 4.1812 & x \\ x & x & x & x & x & x \end{pmatrix}.$$

Si el potencial interno de la neurona ( $ij$ ) supera al valor de umbral, entonces se produce una salida en 1 para esa neurona; al terminar de procesar la imagen se tiene una matriz binaria de salidas  $Y_{ij}$ . El número de iteración en el que sucede la primera activación de cada neurona se guarda en la matriz de tiempos final  $M_{ij}$ .

$$Y_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \underline{1} & 0 & \underline{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, M_{ij} = \begin{pmatrix} 7 & 8 & 7 & 7 & 8 & x \\ 8 & 6 & 7 & 6 & 7 & x \\ 8 & 8 & \underline{7} & 7 & 8 & x \\ 7 & 8 & 8 & 8 & 7 & x \\ 8 & 9 & 8 & 8 & 9 & x \\ x & x & x & x & x & x \end{pmatrix}.$$

La etapa de filtrado selectivo consiste en recorrer las matrices calculando la mediana de los valores de la matriz de tiempos en una ventana de  $5 \times 5$  denotado como *mediana*  $M_{2,2}(5 \times 5)$  y comparar el valor con el correspondiente al pixel central  $M_{2,2}$  de la ventana considerada, cuyos valores respectivos son:

$$\text{mediana } M_{2,2}(5 \times 5) = 8,$$

$$M_{2,2} = 7.$$

Si estas cifras son iguales el valor del pixel correspondiente en la imagen filtrada  $Sfiltrada_{2,2}$  será el promedio de los valores en esta ventana en la imagen ruidosa *promedio*  $O_{2,2}(5 \times 5)$ , en caso contrario el pixel correspondiente tomará el valor de la mediana de la vecindad de  $5 \times 5$  de la imagen con ruido. En este caso no son iguales, se procede a calcular la mediana de la ventana en la imagen con ruido *mediana*  $O_{2,2}(5 \times 5)$  y asignarlo al pixel de la imagen filtrada  $Sfiltrada_{2,2}$ . El proceso termina cuando se ha procesado la imagen completa.

$$\text{mediana } O_{2,2}(5 \times 5) = 138,$$

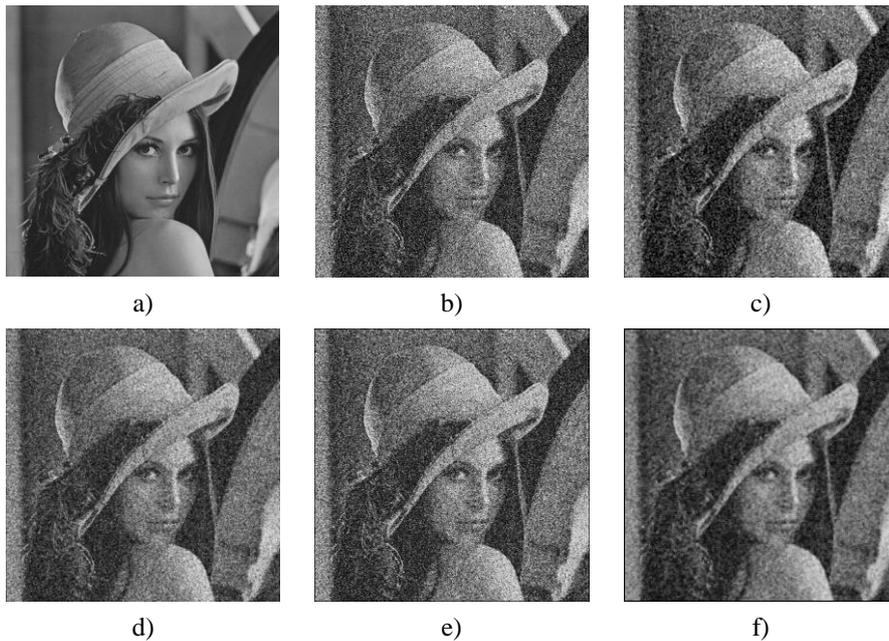
$$Sfiltrada_{2,2} = 138.$$

Se realizó la comparación del método propuesto con los filtros de mediana, Wiener y el basado en PCNNI (Fig. 4); la capacidad de suavizado del filtro de mediana puede ser empleado para disminuir el ruido gaussiano que genera valores extremos en los pixeles, no obstante no se debe aplicar este filtro de manera uniforme pues la imagen

se verá afectada en detalles y bordes, mientras que para otras regiones se debe tomar en cuenta la información presente en los píxeles por medio de un suavizado más fino como es el generado por el operador promedio.

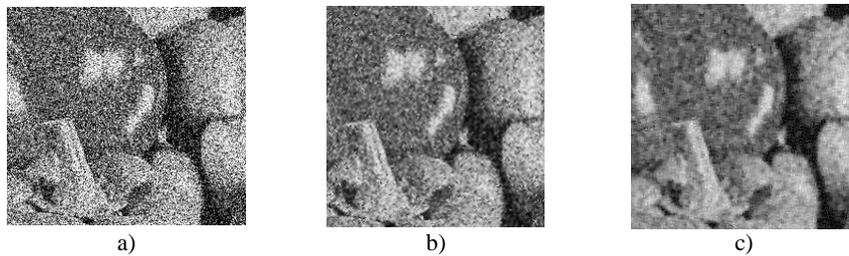
El análisis cualitativo de la imagen ampliada hace evidente que el filtro *Wiener* mantiene un alto grado de ruido aunque preserva los bordes (Fig. 5).

Por otro lado la técnica basada en PCNNNI conserva los detalles finos, pero su capacidad de supresión del ruido no es efectiva (Fig. 6).



**Fig. 4.** Comparación entre técnicas de filtrado de ruido gaussiano a) original sin ruido, b) con ruido gaussiano var. 0.08, c) mediana, d) *Wiener*, e) PCNNNI y f) ICM-TM

El análisis cuantitativo del PSNR (Fig. 7), permite mostrar que el desempeño del ICM-TM es superior por 2dB a los métodos tradicionales de filtrado y a la técnica basada en PCNNNI para imágenes donde el ruido gaussiano tiene varianza 0.07, mostrando este buen desempeño cuando la varianza del ruido es más baja, hasta 0.01 cuando el ICM-TM supera a los métodos clásicos por 1dB. Con varianzas menores el desempeño de la técnica ya no es superior.



**Fig. 5.** a) Imagen con ruido gaussiano var. 0.08, b) filtrada con *Wiener*, c) filtrada con ICM-TM

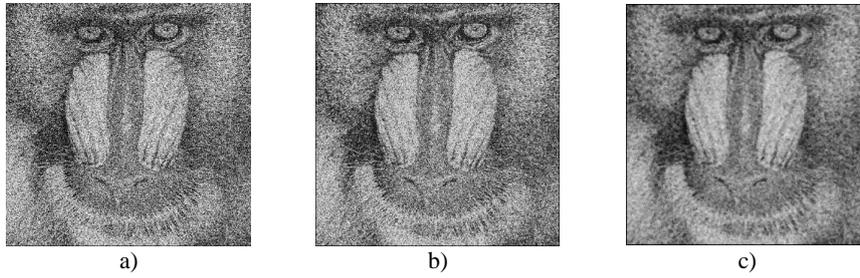


Fig. 6. a) Imagen con ruido gaussiano var. 0.09, b) filtrada con PCNNI y c) con ICM-TM

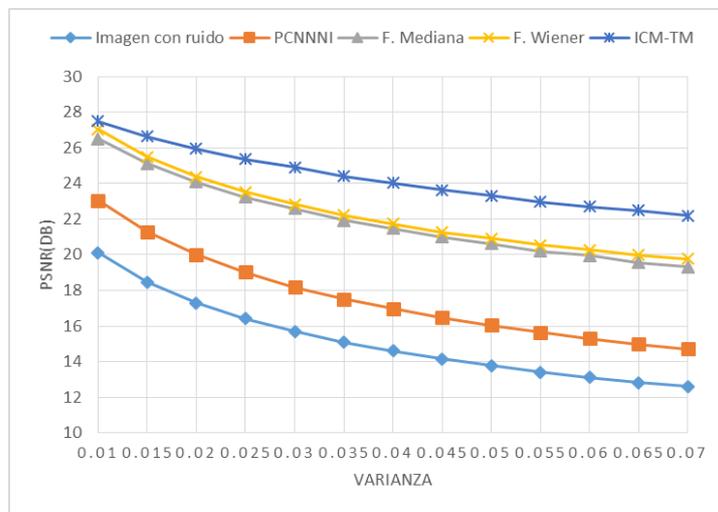


Fig. 7. Curvas de desempeño en PSNR de los filtros de ruido Gaussiano

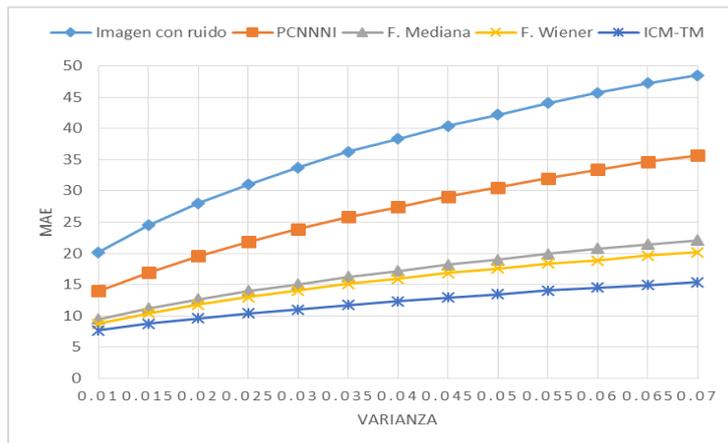


Fig. 8. Curvas de desempeño en MAE de los filtros de ruido Gaussiano

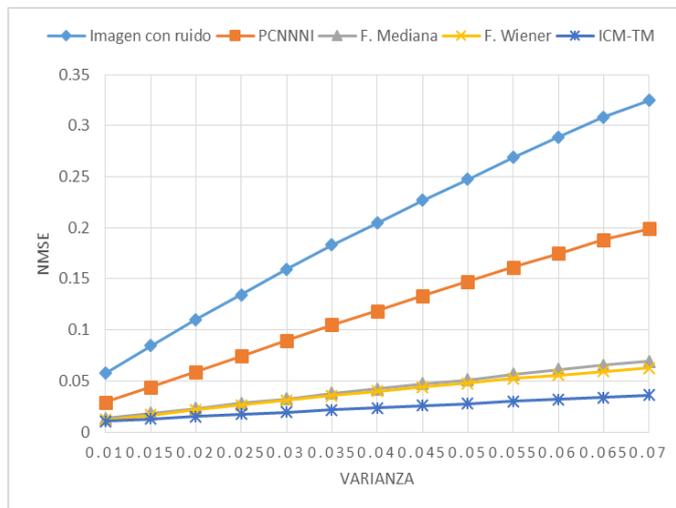


Fig. 9. Curvas de desempeño en NMSE de los filtros de ruido Gaussiano

En cuanto al análisis de los datos reflejados en el MAE se observa que la mejora de la imagen en cuanto a detalles finos es de al menos 30 puntos con respecto a la imagen original en el mejor de los casos y por 10 puntos en el peor (Fig. 8).

Mediante el análisis del NMSE (Fig. 9), se observa que ICM-TM tiende a minimizar esta métrica, lo cual muestra que es un mejor método de filtrado, seguido por el filtro Wiener y el filtro de mediana, donde la diferencia con el método de PCNNNI es notable por 0.3 en el mejor de los casos. Conforme la varianza del ruido aumenta el error de la imagen contaminada y la capacidad de recuperación de los filtros disminuye.

Tabla 1. Desempeño de los filtros de ruido Gaussiano

Imagen con ruido Gaussiano var.	Imagen original	PCNNNI		F. Mediana		F. Wiener		ICM-TM	
	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR
<b>Lena 0.08</b>	12.15	0.224	14.22	0.08	18.7	0.069	19.31	0.041	<b>21.68</b>
<b>Lena 0.09</b>	11.8	0.245	13.83	0.088	18.29	0.075	18.97	0.044	<b>21.36</b>
<b>Baboon 0.08</b>	11.93	0.143	13.9	0.064	17.39	0.05	18.47	0.045	<b>18.9</b>
<b>Baboon 0.09</b>	11.58	0.155	13.53	0.069	17.05	0.053	18.21	0.048	<b>18.66</b>
<b>Peppers 0.08</b>	12.15	0.14	14.27	0.05	18.76	0.044	19.31	0.024	<b>21.97</b>
<b>Peppers 0.09</b>	11.76	0.154	13.85	0.056	18.26	0.048	18.91	0.027	<b>21.52</b>

En cuanto al tiempo el promedio para PCNNNI fue de 160s, para el filtro de Mediana de 0.028s, para el filtro Wiener de 0.036s y para ICM-TM de 20s, por lo que la complejidad del método es acorde con el tiempo de procesamiento aunque éste puede ser mejorado mediante procesamiento en paralelo.

En general para varianzas de ruido gaussiano mayores a 0.01 el desempeño de ICM-TM es superior a PCNNNI, al filtro de mediana y al filtro Wiener para distintas

imágenes contaminadas con ruido gaussiano como muestran los datos de NMSE y PSNR de la Tabla 1 para distintos valores de ruido gaussiano con diversas imágenes.

## 7. Conclusiones y trabajo futuro

Los resultados preliminares del método de filtrado ICM-TM son que supera los filtros tradicionales como filtro *Wiener* en promedio por 1.82 dB según PSNR y en 14% en NMSE y MAE, además de que relaciona una vecindad de píxeles y no requiere cálculos estadísticos ni complejos, del mismo modo aprovecha las características de los filtros promedio y mediana al aplicarlos selectivamente.

Por otro lado ICM-TM supera a PCNN [14] en promedio por 6.75dB, y requiere 80% menos tiempo de procesamiento, además no se realiza un cambio del rango dinámico de la imagen, se considera la relación entre neuronas vecinas y se tienen que ajustar menos parámetros.

El trabajo futuro se centrará en la exploración de PCNN con variantes y la elección de los coeficientes de ajuste por medio de algoritmos evolutivos, así como en la búsqueda de un operador que permita un mejor ajuste de los valores de la imagen para preservar mejor los detalles, el estudio profundo de la matriz de tiempos y tomando en cuenta las frecuencias para la localización de los píxeles que guardan mayor información de la imagen discriminando aquellos que presentan mayor grado de ruido. También se pretende estudiar el comportamiento del filtro como sistema dinámico y su desempeño sobre algunas imágenes artificiales que se enfoca a características específicas como bordes y líneas. Comparar su desempeño con otro tipo de filtros como el direccional o filtros difusos y ampliar el conjunto de imágenes de prueba.

**Agradecimientos.** Los autores agradecen al CENIDET y al IPN por el apoyo económico para la realización de la presente investigación con ayuda de los siguientes fondos: SIP-IPN 20161126, y CONACYT en el marco de los proyectos 155014 de la convocatoria de Investigación básica y 65 en el marco de la convocatoria de Fronteras de la Ciencia 2015. Estela Ortiz agradece al CONACYT por la beca concedida para la realización de sus estudios de maestría.

## Referencias

1. Bovik, A.: The Essential Guide to Image Processing. Image. Elsevier Inc. (2009)
2. González, R.C.: Digital Image Processing. 2nd Ed. Prentice Hall (2002)
3. Kaur, D.: Remove Noise Effects From Degraded Document Images Using Matlab Algorithm. International Journal Of Engineering Sciences & Research Technology, Vol. 4, No. 9, pp. 544–549 (2015)
4. Kang, D., Lim, H.: Efficient noise reduction in images using directional modified sigma filter. In: Springer Science Business Media, New York, pp. 580–592 (2013)
5. Panetta, K., Bao, L., Agaian, S.: Sequence-to-Sequence Similarity Based Filter for Image Denoising. IEEE Sensors Journal (2016)
6. Liu, J., Wang, Y., Su, K., He, W.: Image denoising with multidirectional shrinkage in directionlet domain. Signal Processing, Vol. 125, pp. 64–78 (2016)

7. Kumar A., Singh, B.: Alexander Fractional Integral Filtering Of Wavelet Coefficients For Image Denoising. *Signal & Image Processing : An International Journal (SIPIJ)*, Vol. 6, No. 3, pp. 43–54 (2015)
8. Lindblad, T., Kinser, J.M.: *Image processing using pulse-coupled neural networks*. Springer, 2nd Ed. (2005)
9. Eckhorn R., Reitboeck, H.J., Arndt, M.: Feature linking via synchronization among distributed assemblies: simulation of results from cat cortex. *Neural Computation*, pp. 293–307 (1990)
10. Ranganath, H.S., Kuntimad, G., Johnson, J.L.: Pulse coupled neural networks for image processing. In: *Proceedings of IEEE Southeast Conference, Raleigh*, pp. 26–29 (1995)
11. Ma, Y.D., Shi, F., Li, L.: Gaussian noise filter based on PCNN. In: *Proceedings of 2003 International Conference on Neural Networks and Signal Processing, Nanjing*, pp. 14–17 (2003)
12. Ma, Y.D., Lin, D.M., Zhang, B.D.: A novel algorithm of image Gaussian noise filtering based on PCNN time matrix. In: *Proceedings of IEEE International Conference on Signal Processing and Communication, Dubai*, pp. 24–27 (2007)
13. Lui, C., Zhang, Z.: Sonar images de-noising based on pulse coupled neural networks. In: *Congress on Image and Signal Processing*, pp. 403–406 (2008)
14. Yuan-yuan, C., Hai-yan, L., Xin-ling, S., Jian-hua, C.: A new method of denoising mixed noise using Limited Grayscale Pulsed Couple Neural Network. In: *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, pp. 1410–1413 (2011)
15. Ma, Y.D., Zhan, K., Wang, Z.: *Applications of pulse-coupled neural networks*. Springer, pp. 11–26 (2010)
16. Johnson, J.L., Padgett, M.L.: PCNN model and applications. *IEEE Transactions on Neural Networks*, Vol. 10, pp. 480–498 (1999)